Title of the Invention

PACKET SWITCHING SYSTEM

Background of the Invention

The present invention relates to a packet switching system having an input buffer and an output buffer (referred to as input/output buffer packet switching system, hereinafter), particularly to the packet switching system adopting an arbiter system.

Conventional input/output buffer packet switching system, which has a First-In First-Out (FIFO) memory for each input line, has a disadvantage where if plural packets inputted from plural input lines converge at a predetermined output path, “Head Of Line (HOL) blocking” is caused, suffering from only 58.6% throughput of data transfer. To avoid the HOL blocking, there provides a well known method that has Virtual output Queue (VoQ) for each output path at an input buffer.

The input/output buffer packet switching system, because of a crossbar switch which has no buffer, adopts the way to arbitrate between the VoQs of input buffers, so as to prevent data on the crossbar switch from being converged. The arbitration is performed for deciding a combination of an input port and an output port to give the grant of transmitting data to some of the VoQs. Accordingly higher throughput of data transfer of the switching system depends on efficient arbitration.

There are two ways of arbitration for deciding a combination of an input port and an output port: one where arbitration is performed using the unit of a fixed-length internal packet into which variable length packet inputted to the switch has been divided; another where, as indicated in USP 09/362,134, arbitration is performed using the unit of a fixed-length container into which plural variable length packets packed. In this way, two units are used on a switching processing; one unit of an internal packet having a small fixed length and another unit of a container having a large fixed length.

As regards a unit used on the processing in the switching system of the invention, the volume of data per arbitration is defined as one segment. Note that hereafter the term "segment" is also defined as generic term of data to be treated inside the switch, data such as an internal packet, a packet, and a cell.

As conventional arbitration, the following three methods are proposed:

First is a method of selecting a sending queue on Round Robin by taking whether segment of VoQ exists as a parameter, disclosed in "A Study of structuring a Large Capacity Packet

Switching Systems," Koji WAKAYAMA, et al., SHINGAKUGIHOU SSE98-160, and also disclosed in JP-A-2000-78148;
Second is a method of selecting sending queues by taking a waiting time of the segment in VoQ as a parameter, disclosed in "A Study of Scheduling an Input Buffer Switch and Trial manufacture thereof," Toshiyuki SUDO, et al., SHINGAKUGIHOU SSE99-118;
Third is a method of selecting output data queue by taking the length of VoQ as a parameter, disclosed in "A Proposal of Balanced Packet Scheduling Algorithm and Performance Evaluation," SHINGAKUGIHOU SSE96-56.

Each of the three methods has a problem caused when unbalanced load is given to the switch. Referring to Fig. 16 which is a conceptual view of an input port 30 of a 4 × 4 switch, we will explain the problems below. In the figure, reference numbers 31-1, -2, -3, and -4 are VoQs, each being directed for its output path; a quadrangle in each VoQ represents a segment; the 31-1 shows traffic which has a higher load than other queue; and the 31-4 shows traffic which has a lower load than other queue.

In the first method of selecting a sending queue on Round Robin by taking whether segment of VoQ exists as a parameter, if unbalanced loads are given to the switch, the Round Robin that equitably reads out segments from all VoQs permits the VoQ having low loads to transmit segments under no influence of the VoQ that has high load and is just reading out the segment. However, queues of a VoQ having high load is liable to be long, causing its delay time in segment transmission inside the switch to be longer than those of other VoQs. As shown in the figure, this brings about overflow of segments at VoQ 31-1 having high load, and might abandon a segment 32A.

In the second method of selecting sending queues by taking a waiting time of VoQ of the segment as a parameter, in the same way as the first method, queues of a VoQ having high load is liable to be long. This causes a waiting time of the segment in the VoQ to be long. The method that takes a waiting time of the segment in VoQ as a parameter transmits a segment having a long waiting time in a priority manner. Under the condition, management of the waiting time of all segments needs a lot of counters, so the method is not practicable.

Therefore, in most case, the second method counts the waiting time from when the segment has arrived at the top of the queue. Specifically, it adds 1 to the counter of the VoQ when the segment in the VoQ is not transmitted during one arbitration while resets the counter when the segment is transmitted during one arbitration. This treats both VoQ having high load and VoQ having low load in the same

condition, because the counter resets once the segments of the VoQ which even has high load is transmitted.
Thus, the delay time of the VoQ to which high load is given under the unbalanced load finally becomes large, and thus this might abandon the overflow segments of the buffer.

In the third method of selecting sending queues by taking the length of the queue as a parameter, the segment can be effectively read out from VoQ having high load under above-said condition where unbalanced load is given to the switch. Thus, the delay time of the segment of VoQ having high load comes to be small, also preventing a buffer from overflowing. The method, however, suffers from a phenomenon called starvation in which grant is not given to a segment 32B of VoQ 31-4 that has low load.

Accordingly, the method might deteriorate qualities in packets, especially in both packets of voice data, which is required never to be late in data transfer, and packets of important data, which is required never to be abandoned in data transfer, both for keeping good quality in data.

Summary of the Invention

The invention aims to provide a packet switching system that arbitrates between the VoQs to decide a combination of an input port and an output port, and thereby granting transmitting data to some of the VoQs by taking both an interval in sending a segment from VoQ and queue length of VoQ as parameters.

According to one aspect of the invention we provide the packet switching system having: a queue length manager for managing the volume of segments queued in each VoQ per input line; an output data interval manager for managing an output data interval of the segment of each VoQ; and an arbiter-request (ARB-REQ) generator for allocating level of transmission to the VoQs according to information received from the queue length manager and the output data interval manager, wherein arbitration is performed on the level assigned each VoQ so as to determine which VoQs will be sent.

According to another aspect of the invention we provide the packet switching system having: means for putting segment transfer interval prior to queue length in arbitration so as to determine VoQ level; and means for putting queue length prior to segment transfer interval in arbitration so as to determine VoQ level;

Brief Description of the Drawings

FIG. 1 is a block diagram for explaining one embodiment of a packet switching system of the present invention.

FIG. 2 is a block diagram of the structure of an ARB-REQ generator 13 and a VoQ controller 12 of FIG. 1.

FIG. 3 is one example of a level assignment matrix for assigning a level to VoQ by the ARB-REQ generator of FIG. 13.

FIG. 4 is one example (prioritizing output data interval) of the level assignment matrix for assigning the level to VoQ by the ARB-REQ generator of FIG. 1.

FIG. 5 is one example (prioritizing queue length) of the level assigning matrix for assigning the level to VoQ by ARB-REQ generator of FIG. 1.

FIG. 6 is a block diagram of an example of the structure of an arbiter 14 of FIG. 1.

FIG. 7 is a view for explaining a concept of a VoQ level matrix of the present invention.

FIG. 8 is a view for explaining a concept of a tournament of the present invention.

FIG. 9 is a table of a combination of win and defeat of the tournament for every input and every output of the present invention.

FIG. 10 is a view for explaining a concept of a level reassignment table of the present invention.

FIG. 11 is a flowchart for explaining an algorithm which the arbiter 14 of FIG. 1 performs.

FIG. 12 illustrates processing of the arbiter 14 of FIG. 1.

FIG. 13 is a block diagram of a total structure of another embodiment of a packet switching system of the present invention.

FIG. 14 is a graph of indicating 99 % delay of the conventional arbiter and an arbiter of the present invention.

FIG. 15 is a graph of indicating queue length distribution of the conventional arbiter and the arbiter of the present invention when an unbalanced load is given.

FIG. 16 is a view for explaining a concept of an overflow of VoQ to which high load traffic is given and starvation of VoQ to which low load traffic is given.

Detailed Description of the preferred embodiments

FIG. 1 illustrates one embodiment of a packet switching system of the present invention. In the system, ARB-REQ information is transmitted to an arbiter through a separate line 18 which is different from a data line for connecting VoQ 11 and Crossbar Switch 19.

An input line processor 16-i (i=1 to n) extracts address information of a packet by analyzing a packet header of a packet which is input from an input line 103-i. An input buffer 10-i includes n VoQ11-i to an output port correspondence. An input processor 16-i gives address information which has been extracted to a VoQ controller 12.

The VoQ controller 12-i gives an indication to an input buffer 10-i so as to write the packet in VoQ corresponding to the output corresponding to address information. In this way, the packet is written in VoQ 11-i which has been designated.

The VoQ controller 12-i manages, per VoQ 11-i, information between the volume of queuing segments and a waiting time (that is to say, an output data interval of each segment) of the segment at the top of a queue buffer. An ARB-REQ generator 13-i assigns each level to each VoQ according to those of information.

Each VoQ level is collected to an arbiter 14 by a signal line 18 during one arbitration period. It is decided by the arbiter 14 to give grant to which of VoQ according to its information.

Grant information is transmitted to each of VoQ controllers 12-i(i=1, ⋯, n) as ARB-ACK by a signal line 15, and, at the same time, the arbiter reflects the results of its arbitration to the structure of a path inside a crossbar switch. The VoQ controller 12-i informs it to the input buffer 10 that which of VoQ should send a segment according to its ARB-ACK information.

The segment transmitted from the input buffer 10-i is switched by a crossbar switch 19 and then transmitted to an output line processor 17-i. The output line processor 17-i restructures the packet from the segment which has been received from the crossbar switch 19 and then sends it to an output line 104-i.

With reference to FIG. 2, the structures of the VoQ controller 12 and the ARB-REQ generator 13 are described in detail. At this point, FIG. 2 employs one example of the case of a 4 × 4 switch size for simple explanation.

The header of the packet which has been processed in the header analyzer 401 inside the input processor 16 is transmitted to a write address (WA) generator 403 in the VoQ controller 12, managed to write an input packet to which VoQ11 by the WA generator 403. The WA generator 403 sends memory address information to the input buffer 10 by way of a WA control signal line 412, and gives an indication of writing the packet to VoQ 11 corresponding to a destination. At the same time, the WA generator 403 transmits information of the packet which has been written in the input buffer 10 to a queue length manager 405 and an output data interval manager 406. The queue length manager 405 has a queue length counter 410 corresponding to each of all VoQs inside the input buffer 10.

FIG. 2 gives the case of a 4 × 4 switch as an example. Since four VoQs exist in the input buffer 10, the queue length manager has four queue length counters 410. The queue length manager 405 increases the length of the segment of the input packet to the numeric value of the queue length counter 410 for the current length of the queue. The output data interval manager 406 has an output data interval counter 411 corresponding to one or more VoQs inside the input buffer 10. The output data interval manager 406 does nothing to the VoQ in which the packet has been input in the case where the segment has already existed. If the segment has not existed, the output data interval manager 406 gives an indication to the output data interval counter 411 corresponding to the VoQ so as to add 1 to the numeric value per arbitration period, and manages the output data interval time. In other words, the numerical value which the output data interval counter 411 shows indicate that how long the segment has not been transmitted from corresponding VoQs.

A read address (RA) generator 404, according to ARB-ACK information transmitted from the signal line 15, transmits, through a signal line 413, information to send from which VoQ the segment to the input buffer. At the same time, the read address (RA) generator 404 transmits information of the segment which is read out from the input buffer 10 to the queue length manager 405 and the output data interval manager 406 as well. The queue length manager 405 decreases the queue length counter 410 corresponding to VoQ which has transmitted out the segment to the crossbar switch 19. Further, the output data interval manager 406 resets the value of the output data interval counter 411 corresponding to the VoQ.

Information of the queue length manager 405 and of the output data interval manager 406 is transmitted to the ARB-REQ generator 13 by way of a signal line 414. The ARB-REQ generator 13 has an ARB-REQ generating part 409 corresponding to each queue inside the input buffer 10. Respective ARB-REQ generating parts 409-1 to -4 assign some level to the corresponding queue according to information received from the ARB-REQ generator 13. When the level is assigned to the queue, a VoQ level assignment matrix 416 is referred. For the VoQ level assignment matrix 416, it is possible for a user to tune the arbiter in accordance with the characteristics of the traffics which are input to the node thereof. The level of each VoQ which has been created in the ARB-REO generator is transmitted to the arbiter 14 by way of the signal line 18.

FIG. 3 shows one embodiment of the VoQ level assignment matrix 416. The level assignment matrix has a segment transfer interval 71 in a horizontal axis and the number of segments queued in VoQ in a vertical axis 72. The longer an output data interval time is and the more the number of segments queued in VoQ is, the bigger the level assigned to VoQ is. The level assignment matrix is calculated from a queue length (the number of segment in VoQ) and the segment transfer interval.

By assigning the level to the queue in this way, it comes to be possible to send within a delay time for setting arbitrarily the packet which has entered into the switch.

The time for sending the packet from the queue of VoQ having a high load is longer than that of VoQ having a low load. Thus, a high level is assigned to VoQ in which the number of segments stored in the queue is big, and thus grant is given thereto in a priority manner. In other words, this level indicates information of the degree of priority in obtaining grant.

Since the packet is transmitted within a delay time which is set arbitrarily by way of VoQ having low load, it functions to increase the VoQ level if the transmitting interval becomes long.

The VoQ level L is obtained from the following expression.

$$L = \frac{1}{\ln\left(\frac{M - a \cdot t}{b \cdot s} \times e\right)} \times 15$$

where, M: time out, t: output segment interval, s: the number of segments at the present time, a: output data coefficient, b: queue length coefficient.

When (M-a·t)/b·s ≤ 1 is attained, the VoQ level attains the maximum value. When the VoQ level has reached the maximum value, it is indicated that the VoQ thereof has reached to such a condition as having a high possibility of obtaining a grant.

M is a value which is defined from both a delay time that can arbitrarily be set by the switch and one arbitration time. M is also a value for deciding both the maximum value of segment transfer interval 71 in level assignment matrix and the maximum value of the number of segments 72 queued in VoQ. Where T is the delay time which is required by the switch and determined arbitrarily, ta is a time for one arbitration time, n is the number of input ports of the switch. M can be obtained by the following expression.

$$M = \frac{T}{t_a} - n$$

T/ta can define the number of times of performing arbitration during the delay time T which can be defined arbitrarily. On the assumption that the levels of all VoQs attain the maximum values at the same time, until the grant is given, VoQ, to which the time for maximum n arbitration will be waited, appears. Therefore, in order to transmit the segment within the delay time which is arbitrarily defined, even if VoQ has only one segment, a VOQ level takes maximum, when the output data interval becomes T/ta-n.

As shown from the level assignment matrix shown in FIG. 3, since the VoQ level becomes high as the output data interval turns to be large in VoQ having only one segment, it is not necessarily concluded that the segments are not transmitted until the maximum delay time is requested.

The level assignment matrix in FIG. 3 is limited to time out M = 20, a = 1, and b = 1. By changing an output data interval coefficient and a queue length coefficient *b*, it is possible for them to be changed to arbitration which regards the output data interval as important and arbitration which regards a queue length as important.

When the packet switching system of the present invention is employed in a place where, for example, a lot of voice data required never to be late in data transfer, should be processed, setting is changed so as to suppress the delay time as much as possible where such data are queued in VoQ having low load traffics.

Concretely, setting the value of the output data interval coefficient *a* to 1 or more permits the VoQ in which the segments are not yet filled to get a large level within a short output data interval. FIG. 4 shows the condition of the VoQ level assignment matrix at the time of defining the value of the output data interval coefficient *a* as 2. When the level of the matrix in FIG. 4 is compared with the level assignment matrix of FIG. 3, the VoQ level already becomes large when the output data interval of the segment is small. Thus, it is also possible for VoQ having the low load to transmit the segment in a short delay time.

On the contrary, when the packet switching system of the present invention is employed in a place where a lot of data required never to be abandoned in data transfer, should be processed, though the delay time of VoQ having the low load traffic lengthens slightly, it is preferable to suppress an overflow of a buffer by outputting the segments from VoQ having the high load traffics in a priority manner. In such a case, by defining the value of the queue length coefficient *b* as the value greater than 1, it is permitted to assign the VoQ level which acts in response sensitively to a change of the length of the queue. When the high load is applied, the length of the queue turns to be long. FIG. 5 exemplifies the condition of the level assignment matrix at the time of assigning the queue length coefficient as 2. When the level assignment matrix is compared with the level assignment matrix of FIG. 3, large levels are found in places in which the number of the segments of VoQ is small.

Therefore, for VoQ in which the length of the queue becomes longer, it is possible to prevent VoQ in which the length of the queue becomes longer from the buffer's overflowing by making it easy to give grant by giving a larger level as soon as earlier.

Further, where the output data interval is not considered at all and it is desired that arbitration is performed using only the length of the queue, it comes to be possible by defining the value of the output data interval coefficient *a* as 0.

All of the VoQ levels calculated by a numeric expression 1 are rounded off and they are expressed in the level assignment matrix as integers. Further, when $M-a \cdot t < b \cdot s$ or $M \leq a \cdot t$ is attained, they become the values other than $0 < L \leq 15$. However, when such a situation occurs, since it is expressed that the VoQ level already exceeds the maximum level 15, the level 15 is given to VoQ to which the values other than $0 < L \leq 15$ are given by this expression.

Information of the level per VoQ is collected to arbiter 14 from each ARB-REQ generator.

FIG. 6 is a block diagram of an embodiment of the arbiter 14. In all VoQ levels collector 121, information of the level of all VoQs is collected.

FIG. 7 expresses the condition of all VoQs to which requests have been made as of arbitrary points of time. The column of the matrix expresses an output line number 131, and the row thereof expresses an input line number 132. For example, in the case where an input line number is 1 and an output line number is 1, the level, which is assigned to VoQ 11-1-1 of an input buffer 10-1 of FIG. 1, is stored. Further, a "O" 133 of the matrix has the same VoQ level and the level smaller than "O" is assigned to an empty portion.

In this condition, giving grant to VoQ having an output line number 4 in an input line number 2 and VoQ having an output line number 2 in an input line number 4 obtain the best combination of inputs and outputs. In order to give grant to VoQ efficiently all the time, a tournament for each of inputs and a tournament for each of outputs are performed.

FIG. 8 is a view of a concept of tournament processing. In FIG. 8, numerals represent the level of VoQ which is a member of the same input line number or output line number. In FIG. 8, though there are two VoQs having the same levels, in such a case, it is made not to win either one but to win all VoQs having the same levels. This is to give grant efficiently as shown in FIG. 7.

This tournament processing is performed for each of input line directions and for each of output line directions in an every input line tournament processor 122-1 and an every output line tournament processor 122-2, and then VoQ is selected, which has the highest VoQ level (the request of transmitting is the highest among their line numbers) among them.

As the result of the tournament, a win/defeat combination 141 of FIG. 9 is capable of being considered for an input line direction and an output line direction. The level reassignment part 124 evaluates information of each VoQ level by reducing to 4 levels, 0 to 3, according to a level reassignment table 61 of FIG. 10.

VoQs which have been reevaluated into 4 levels, 0 to 3, are picked up sequentially from VoQs of the level 3 in a selector 125 of VoQ having the same level. Grant is given to VoQ which has been picked up herein by a Round Robin selection in a grant assignment part 126.

Since it is not possible to give grant in the same arbitration period from both VoQ to which grant is given and VoQ which is a member of the same input line number or the same output line number, grant is taken away in a grant deprival part 127.

Information of VoQ, grant of which has been deprived, is notified to all VoQ levels collector 121 by way of a signal line 123. From its information, the tournament is performed once more among VoQs, grant of which has not been deprived at all, and then the levels thereof are reevaluated. Then, in the same way as the aforementioned description, VoQ having level 3 is picked up by way of a same level VoQ selector 125, and then grant is given to the VoQ by a grant assignment part through the Round Robin selection. By repeating such a repetitive operation, it is possible to create the best combination of input line and output line.

Since it is not possible to give grant at the same time in the same arbitration period from VoQs that are members of the same input line number or the same output line number as VoQs having grant in the aforementioned process, the grant deprival part 127 deprives grant from the VoQs. Information of VoQ, grant of which has been deprived, is notified to the same level VoQ selector 125 by way of a signal line 129. The same level VoQ selector 125 picks up VoQ having level 2 still having grant and then gives grant by the grant assignment part through the Round Robin selection.

Then, by way of the same process as giving grant to VoQ having level 2, grant is given to VoQ having level 1. Grant is also given to VoQ with the level 0 having grant to be transmitted next.

Grant information is changed to ARB-ACK information by an ARB-ACK generator 128, and then transmitted to the RA generator 404 of the VoQ controller 12 by way of the signal line 15. The RA generator 404 transmits the segment-transmitting signal 413 to VoQ, according to ARB-ACK information. At the same time, VoQ information for sending the segment is notified to the queue length manager

405 and the output data interval manager 406. The queue length manager 405 decreases the number of transmitted segments from the value of the queue length counter 410 which manages the number of segments of VoQ to which grant has been given. In the output data interval manager 406, the value of the sending out interval counter 411, which manages the output data interval of VoQ to which grant has been given, is reset.

FIG. 11 is a flowchart of a sequential process of the aforementioned arbiter 14. The tournament is performed to the VoQ levels which have been collected from each of the input buffers with respect to the input and the output (S81). The VoQ levels thereof are reevaluated using the levels of 3 to 0 in order of highly requested output data for each VoQ (S82-1 to -4). First of all, VoQs of the level 3 are picked up by way of the same level VoQ selector 125 (S83). Grant is given to them through the Round Robin (S83-1) (it may be considered to adopt 2DRR (Mamoru TAKAHASHI, et al., "Improvement of Packet-Priority-considered-Packet-Switch having Input Queue corresponding to each Output Port," SHINGAKUGIHOU SSE97-13) and the like, which has a pointer in order to maintain the state of being equal as this Round Robin). Since it is not possible to transmit the segment at the same time as the same arbitration period from VoQ for the same input and the same output as VoQ to which VoQ has given, grant of VoQ is deprived (S83-2).

In the present embodiment, in order to improve efficiency of the arbiter, not to give grant to VoQ having level 2, but the level of VoQ, to which grant is not given, is back to the level of 0 to 15 before reevaluating, the tournament is performed once more, and then reevaluation of the level (S84) is performed. The more this process is repeated, the more the combination of the queues selected by arbitration is closed to the most suitable one.

As a result of reevaluating, grant is given to VoQ having level 3 (S83-1). Grant of VoQ, which has the same combination of input lines and output lines as the VoQ obtaining grant, is cancelled (S83-2).

Subsequently, VoQ having level 2 is picked up (S85), grant is given (S85-1), and then grant of VoQ which has the same combination of input lines and output lines as VoQ to which grant has given is canceled (S85-2).

Subsequently, VoQ having level 1 is picked up (S86), grant is given (S86-1), and then grant of VoQ which has the same combination of input lines and output lines as VoQ to which grant has given is canceled (S86-2).

At last, if there is VoQ having grant at the level 0, grant is given by way of the Round Robin selection (S87). In this way, it is ended to give grant to the combination of all of the inputs and the outputs, and then the process of arbitration has been finished.

FIG. 12 is the results of processing in accordance with the flowchart of FIG. 11. This figure indicates arbitration of the 4 × 4 switch. It is possible to express the VoQ levels, which have been collected in all VoQ levels collector 121, visually in a matrix 21 way. The row of the matrix indicates the input line number, and the column indicates the output line number. This matrix shows that, for example, in the matrix 21, the VoQ level for the input line number 3 and the output line number 1 is 10.

The tournament of FIG. 8 is performed for each of the input lines and the output lines, and thereafter aforementioned reevaluation is performed in FIG. 10. A matrix 22 indicates the results thereof.

The matrix 23 shows that VoQ having level 3 has been selected from the matrix 22 and grant has been given thereto by way of the Round Robin selection. Since grant cannot been given to VoQ which has the same input line number or output line number as VoQ to which grant has been given at the same arbitration period, grant is deprived therefrom. The symbol "x" of the matrix 24 indicates that grant has been deprived.

Next, The level of VoQ to which grant has not yet been given is back to the level of VoQ which has been created in the ARB-REQ generator. A matrix 25 indicates the matrix which has already been converted. Once more, the tournament and reevaluation of the levels are performed.

A matrix 26 indicates the result thereof. In the matrix 26, grant is given to VoQ having level 3. At this point, grant is given to VoQ having the input line number 2 and the output input number 4.

A matrix 27 indicates that grant has been given thereto. Grant is deprived from VoQ which has the same input line number and the output line number as VoQ to which grant has been given.

A matrix 28 indicates that grant has been deprived therefrom. Next, grant is given to VoQ having level 2.

A matrix 29 indicates that grant has been given thereto.

In the case of being explained in FIG. 6, since grant has been given to all of the combinations of inputs and outputs in accordance with the aforementioned processes, a series of arbitration has terminated. For the case other than that, there may be a case where a process of giving grant to VoQ having level 1 and the level 0 is required. In such a case, grant is given in accordance with the flowchart of FIG. 1.

FIG, 13 is another embodiment of the packet switching system of the present invention. In the embodiment, by giving ARB-REQ information to a header portion of the segment without using another line, it is transmitted to an arbiter via in-channel.

The points different from the packet switching system of FIG. 1 are that ARB-REQ information is once transmitted to an ARB-REQ assignment part 111 by way of a signal line 118 and then to the arbiter 14 by giving ARB-REQ information 114 to the header portion of a segment 113 and that the arbiter 14 is included inside the crossbar switch 19.

In the same way as the embodiment of FIG. 1, ARB-REQ information is collected in the arbiter 14, and then it is decided to give to which VoQ from ARB-REQ information. Then, grant information is assigned to a header 116 of a switched segment 115 as ARB-ACK from the arbiter by way of the signal line 129. Grant information is collected in an ARB-ACK collection part 112, and then transmitted to the VoQ controller 12 by way of a signal line 119. The VoQ controller 12 instructs an input buffer to transmit the segment from which VoQ.

The advantageous point of this method is that it is possible to make simple the structure of a hardware since the number of the signal lines can be decreased because it is not required to prepare the signal line for the arbiter.

FIGS. 14 and 15 show results of the simulation of a queue information management arbiter of the present invention and an arbitration method (referring to Koji

WAKAYAMA, et al., "A Study of structuring a Large Capacity Packet Switching Systems," SHINGAKUGIHOU IN98-160) for performing the Round Robin by judging the presence of the segment of VoQ under the same condition. It is supposed that the condition of the simulation is a 4 × 4 input output crossbar switch having the input line number 4 and the output line number 4.

FIG. 14 shows a distribution graph of an average delay time of the conventional arbiter and the proposed arbiter of the present application at the time of having uniform traffic. The vertical axis 92 is Delay (Segment), and the horizontal axis 91 has Load Rate (%). The higher the load of a line is, the bigger the volume of the delay of the conventional arbiter is. However, it is possible for the proposed arbiter to suppress the delay time from increasing.

Even for uniform traffic, if the load of the line becomes higher, the traffic condition tends to be unbalanced. Therefore the proposed arbiter which takes both output data interval and queue length of VoQ as parameters can suppress the delay time better than the conventional arbiter which takes whether the segment exists as a parameter.

FIG. 15 shows the results of the simulation of the delay distribution of VoQ having low load traffic and VoQ having high load traffic when traffic having the load higher than others is given to one input line among 4 input lines. The vertical axis 102 is Probability (Delay time > d), and the horizontal axis 101 is Delay (segment).

Quadrilateral plots express the delay distribution of the input port having traffic of the higher load, and triangle plots express the delay distribution of the input port having traffic of the lower load.

As shown from the delay distribution of VoQ having traffic of the higher load, the proposed arbiter suppresses the delay time better than the conventional type arbiter. The length of the queue of VoQ having traffic of the higher traffic is longer than that of other VoQs. The proposed type of performing arbitration taking the length of the queue as a parameter tends to give much grant to VoQ having long queue length. This can suppress the delay time of VoQ having traffic of the higher load.

On the other hand, though an impact is given to VoQ having traffic of the low load by said effect, the switching system of the present invention can suppress the effect to VoQ having low load traffic because it takes output data interval as a parameter.

Since the length of the queue is managed, it is possible to perform highly effective switching even when unbalanced loads are given to the switch. It is possible to suppress the delay time of VoQ effectively, to which traffic of the high load is given. Further, it is possible to transmit the segment without giving the effect to VoQ to which traffic of the low load is given.

Even for VoQ to which traffic of the low load is given, it is possible for VoQ level to take the maximum level when the time predetermined by a user arrives. Thus, it comes to be easy to remain undefeated in the input direction and the output direction for the tournament of next processing and thus it comes to possible to attain the maximum level when the level is reevaluated. Therefore, it comes to be easy for its VoQ to obtain grant of the segment. Therefore, it comes to be possible to prevent starvation of VoQ to which traffic of the low load is given.

By employing the present invention, it is possible to provide the arbiter capable of managing any of the balanced loads or the unbalanced loads.